

### **Remarks**

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-15, 18 and 20 are pending in the application. Claims 1-15, 18 and 20 are rejected. Claims 16, 17 and 19 have been canceled without prejudice. No claims have been allowed. Claims 1, 7, and 15 are independent.

### ***Telephonic Interview***

Applicant wishes to thank the Examiner for extending a telephonic Examiner Interview on July 25, 2007. Claims 1 and 3 and U.S. Patent No. 5,659,555 to Lee et al. (Lee) were discussed. Although specific agreement was not reached, Applicant now presents reasoned arguments in light of the Examiner's description of Lee.

### ***Previous Drawing Objections***

Applicants respectfully thank Examiner for withdrawing the objections to the drawings.

### **35 U.S.C. § 101 Rejections of Claims 1-6 and 15-20**

The Examiner rejects claims 1-6 and 15-20 on the grounds that the claims recite a software algorithm that does not produce a useful tangible output, thereby failing to provide a tangible and concrete result.

#### **Claims 1-6.**

The Action rejects claim 1 on the grounds that it "only recites software steps and does not produce a useful tangible and concrete final result." [Action, p. 4, ¶9.2] Applicants respectfully disagree with the Examiner's characterization of the claim and relevant law, and believes that the claim in its previous state satisfied 35 USC § 101.

Claim 1 recites: *executing the program under test under test conditions using the stored created strategies that cause the program to have a higher probability to execute through states that correspond to the untested program behavior.* This provides "an improved method for obtaining testing coverage for software applications that behave non-deterministically." [Specification, page 4, lines 5-6.] Executing a program under test in such a manner as to obtain improved testing coverage for

software applications that behave non-deterministically goes beyond simply reciting software steps, and produces a useful, tangible, and concrete result—the well-tested program. Thus, at least for this reason, Applicants respectfully submit that claim 1 is directed to statutory subject matter and request that the rejection under 35 U.S.C. § 101 be withdrawn. Claims 2-6 depend on Claim 1 and at least for that reason should also not be subject to a 35 U.S.C. § 101 rejection.

### **Claims 15-20.**

The Action rejects claim 15 on the grounds that the claim feature “computer readable medium” is only recited in the preamble and is therefore “not required for the life meaning and vitality of the claim limitations.” Applicants respectfully disagree with the Examiner’s characterization of the claim and relevant law, and believe that the claim in their previous state satisfied 35 U.S.C. § 101. Nevertheless, applicants have amended claim 15 in an effort to expedite prosecution. Specifically, the “instructions” are now specifically “*stored on the computer-readable medium.*” As the claim now unambiguously is directed to functional subject matter stored on a computer-readable medium, Applicants respectfully submit that claim 15 is directed to statutory subject matter and request that the rejection under 35 U.S.C. § 101 be withdrawn. Claims 16-20 depend on Claim 15 and at least for that reason should also not be subject to a 35 U.S.C. § 101 rejection.

### **35 U.S.C. § 112 Second Paragraph Rejections of Claim 13**

Claim 13 is said to lack antecedent basis for the phrase “the created strategies.” Applicants respectfully disagree with the Examiner’s characterization of the claim and relevant law, and believe that the claim in their previous state satisfied 35 U.S.C. § 101. Nevertheless, Applicants have amended claim 13 in an effort to expedite prosecution, and to provide explicit antecedent basis for the claim 13 language “the created strategies.” Applicants respectfully request that the rejection be removed.

### **35 U.S.C. § 102 Rejections of Claims 1-20**

The Action rejects claims 1-20 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,659,555 to Lee et al. (Lee). Applicants respectfully submit the claims are allowable over the cited art. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not describe each and every element.

Accordingly, applicants request that all rejections be withdrawn.

### **Claim 1.**

Amended claim 1 recites “receiving a graph of edges and states *representing a program under test*; creating a continuous cycle of edges through the graph *that reaches each edge in the graph at least once*.”

Lee fails to teach or suggest the above claim language.

In support of his position, the Examiner directs Applicants to the following passage in Lee:

In the preferred embodiment, a counter is kept of upgraded external input transitions. If it does not increase for a while then we notice that the test has made no progress. Special measures have to be taken to break out of this situation. The no progress problem could be caused by enforcing priority when choosing the next external input transition to test; the test sequence may trace the composite machine along a loop, and there is no way we can find it directly. One way to break the loop is to relax the priority of the next transition to test temporarily, i.e., select the next external input transition to exercise randomly, disregarding classes. After a few iterations, resume normal test operation. [Lee 12:4-14, emphasis added.]

The Examiner equates the loop, above to the claimed cycle. [Action, p. 7, ¶18.1.]

Lee describes developing protocols to test “communicating FSMs” which does not use a directed graph to perform the testing. Rather, a list of all states and all transitions between the states is generated. Then, with no more fuss, the program under test is run. The decisions about which transitions to test are made when the program is running--not before. When the program is running, a transition between currently reachable states is chosen. If there are transitions that have yet to be tested ( $T_0$ ), one is randomly chosen and tested. If all transitions have been tested at least once, if there are transitions that have only weakly been tested ( $T_1$ —the ending state has not been reached), a weakly tested transition is randomly chosen and tested. Otherwise, an already-tested transition ( $T_2$ ) is tested. The output from the test is then compared to the specification. [See Lee at Fig. 5; pseudo-code at 8:53-9-6; Test Generation Method at 9:8-9:32.]

The number of external input transitions that have been newly tested is kept as a counter value. If the counter value doesn't change for awhile, then an assumption is made that the program could be in a loop. In such cases, to break the loop, the next external state transition is chosen randomly for a few transitions, rather than following the protocol described above. [See Lee at 12:4-14.] Thus, Lee

describes a testing program getting into an unproductive, inadvertent, loop of unknown size. The loop is produced by the testing program choosing a random set of transitions following a specific protocol. Though, the loop could be thought of as a continuous cycle of edges through the graph, it does not *reach each edge in the graph at least once*, where the graph represents *a program under test*. The loop does not produce any newly tested transitions, and so does not reach, at a minimum, any edges leading to transitions that were not previously tested in the program under test. Further, the edges that are reached in the graph by the loop do not “represent a program under test.” At best, they represent a random, unproductive, section of the program under test.

Further, a program using a random protocol to select transitions that produce an unwanted loop is not *creating* a continuous cycle of edges, rather, at most, it is *traveling along* an ad-hoc continuous cycle of edges. The cycle is ad-hoc as it is just a result of a string of decisions made separately at each transition. An ad-hoc loop does not teach or suggest “creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once.”

For these reasons, at least, claim 1 should be allowed. Such action is respectfully requested.

Further, Lee does not teach or suggest the amended language “***creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once; splitting the continuous cycle into discrete sequences that end at edges reaching non-deterministic states uncontrollable by the testing environment.***”

To teach or suggest ***splitting the continuous cycle into discrete sequences that end at edges reaching non-deterministic states uncontrollable by the testing environment***, the Examiner cites Lee at Fig. 5, and its description, which describes the protocol that Lee teaches to choose, in a running testing program, the next external state transition to test. [See description above, and Lee at Fig. 5; pseudo-code at 8:53-9-6; Test Generation Method at 9:8-9:32.]

The claim language requires that the same “continuous cycle of edges through the graph” then be split into discrete sequences. To teach the “continuous cycle of edges,” the Action relies on an inadvertent loop in Lee which does not generate any new state transition tests. The path of this loop is an error, and as such, is not saved or otherwise used. Further, the loop is ad hoc, as it is the result of mostly-random choices made at each transition. As such this ad-hoc set of loop transitions that is not saved is not, and cannot be, *split* into discrete sequences, let alone ***splitting the continuous cycle into discrete sequences that end at edges reaching non-deterministic states uncontrollable by the testing***

*environment.”*

For these further reasons, at least, claim 1 should be allowed. Such action is respectfully requested.

### **Claims 2-6.**

Claims 2-6 ultimately depend on claim 1. Thus, at least for the reasons set forth above with respect to claim 1, claims 2-6 should also be in condition for allowance. These claims also set forth independently patentable combinations of method acts.

### **Claim 7.**

Amended claim 7 recites “a graphing a continuous cycle touching all edges of the abstract state machine and splitting the continuous cycle into discrete sequences that end at non-deterministic states program....” Lee fails to teach or suggest the above claim language.

In support of his position, the Action directs Applicants to the following passage in Lee:

In the preferred embodiment, a counter is kept of upgraded external input transitions. If it does not increase for a while then we notice that the test has made no progress. Special measures have to be taken to break out of this situation. The no progress problem could be caused by enforcing priority when choosing the next external input transition to test; the test sequence may trace the composite machine along a loop, and there is no way we can find it directly. One way to break the loop is to relax the priority of the next transition to test temporarily, i.e., select the next external input transition to exercise randomly, disregarding classes. After a few iterations, resume normal test operation. [Lee 12:4-14, emphasis added.]

The Action equates the loop, above to the claimed cycle. [Action, p. 7, ¶18.1.]

Lee describes developing protocols to test “communicating FSMs” which does not use a directed graph to perform the testing. Rather, a list of all states and all transitions between the states is generated. Then, with no more fuss, the program under test is run. The decisions about which transitions to test are made when the program is running--not before. When the program is running, a transition between currently reachable states is chosen. If there are transitions that have yet to be tested ( $T_0$ ), one is randomly chosen and tested. If all transitions have been tested at least once, if there are transitions that have only weakly been tested ( $T_1$ —the ending state has not been reached) a weakly tested transition is randomly chosen and tested. Otherwise, an already-tested transition ( $T_2$ ) is tested.

The output from the test is then compared to the specification. [See Lee at Fig. 5; pseudo-code at 8:53-9-6; Test Generation Method at 9:8-9:32.]

The number of external input transitions that have been newly tested is kept as a counter value. If the counter value doesn't change for awhile, then an assumption is made that the program could be in a loop. In such cases, to break the loop, the next external state transition is chosen randomly for a few transitions, rather than following the protocol described above. [See Lee at 12:4-14.] Thus, Lee describes a testing program getting into an unproductive, inadvertent, loop of unknown size. The loop is produced by the testing program choosing a random set of transitions following a specific protocol. Though, the loop could be thought of as a continuous cycle of edges through the graph, it does not *graph a continuous cycle touching all edges of the abstract state machine*. The loop does not produce any newly tested transitions, and so does not reach, at a minimum, any edges leading to transitions that were not previously tested in the program under test, and thus, does not touch all edges of the abstract state machine.

Further, a program using a random protocol to select transitions that produce an unwanted loop is not *creating* a continuous cycle of edges, rather, at most, it is *traveling along* an ad-hoc continuous cycle of edges. The cycle is ad-hoc as it is just a result of a string of separate decisions made separately at each transition. An ad-hoc loop does not teach or suggest "creating a continuous cycle of edges through the graph that reaches each edge in the graph at least once."

For these reasons, at least, claim 1 should be allowed. Such action is respectfully requested.

Further, Lee does not teach or suggest "a *graphing a continuous cycle touching all edges of the abstract state machine and splitting the continuous cycle into discrete sequences that end at non-deterministic states program....*."

To teach or suggest *splitting the continuous cycle into discrete sequences that end at edges reaching non-deterministic states*, the Action cites Lee at Fig. 5, and its description, which describes the protocol that Lee teaches to choose, in a running testing program, the next external state transition to test. [See description above, and Lee at Fig. 5; pseudo-code at 8:53-9-6; Test Generation Method at 9:8-9:32.]

The claim language requires that the same "continuous cycle" then be split into discrete sequences. To teach the "continuous cycle of edges" the Action relied on an inadvertent loop in Lee which does not generate any new state transition tests. The path of this loop is an error, and as such, is

not saved or otherwise used. Rather, the loop is ad hoc, as it is the result of mostly-random choices made at each transition. As such this ad-hoc set of loop transitions that is not saved is not, and cannot be, *split* into discrete sequences, let alone *splitting the continuous cycle into discrete sequences.*”

For at least these reasons, claim 7 is in condition for allowance.

### Claims 8-14.

Claims 8-14 ultimately depend on claim 7. Thus, at least for the reasons set forth above with respect to claim 7, claims 8-14 should also be in condition for allowance. These claims also set forth independently patentable combinations of method acts

### Claim 15.

Amended claim 15 recites:

A computer-readable medium having thereon computer-executable instructions comprising:

instructions stored on the computer-readable medium for creating a model of program behavior comprising an abstract state machine with edge transitions;

instructions stored on the computer-readable medium for verifying program behavior;

instructions stored on the computer-readable medium for splitting the model of program behavior into sequences of at least two edge transitions ending at non-deterministic behavior;

instructions stored on the computer-readable medium for determining strategies for the sequences of at least two edge transitions ending at non-deterministic behavior more likely to reach an identified program behavior; and

instructions stored on the computer-readable medium for causing a program to execute behavior corresponding to the strategies for the sequences of at least two edge transitions ending at non-deterministic behavior more likely to reach the identified program behavior.

The Action rejects claim 15 with a general reference to claims 1 and 7, stating: “note the rejection of claims 1 and 7 above. The Instant Claim recites substantially the same limitations as the above-rejected claims and therefore is rejected under the same prior-art teachings.” [Action, page 13.] Applicants respectfully disagree. Claim 15 is not substantially the same as claims 1 and 7. For example, claim 15 recites “instructions stored on the computer-readable medium for determining strategies for the sequences of at least two edge transitions ending at non-deterministic behavior more

likely to reach *an identified program behavior*.”

Initially, it is noted that the Office has not identified where in Lee an alleged anticipatory teaching is to be found. Additionally, Fig. 5 of Lee, cited by the Office Action in support of the contentions that Lee teaches the feature “creating strategies” of claim 1 and “a strategy calculation program” of claim 7 as well as the written disclosure associated with that Fig. 5, are silent as to the subject feature. Still further, Applicant has carefully reviewed the remainder of patent to Lee, and finds no teaching of “determining strategies for the sequences more likely to reach an *identified program behavior*.” Lee does not discuss, e.g., “an identified program behavior.” Thus, for at least this reason, Lee cannot anticipate independent claim 15.

Accordingly, as the Action has failed to make a *prima facie* case against claim 15, favorable reconsideration and withdrawal of the rejection of independent claim 15 under 35 U.S.C. §102 are respectfully requested.

As a separate reason for patentability, Lee fails to teach or suggest the claim 15 features: *wherein determined strategies are determined based on a comparison of edges exiting a deterministic state representing program behavior, and a selected edge has a highest probability of reaching a state representing the identified program behavior.*

As already mentioned, Applicants respectfully note that they cannot locate the feature “the identified program behavior” within the teachings of Lee. The Action rejects the above language (in its rejection of the similar claim 19 language) using three references to Lee: to Fig. 5 and its description, 12:4-14; and 6:50-55; each of which will be dealt with in turn.

Lee: Fig. 5 and its description. This describes running a program under test using a guided walk. When a program is running, the next transition to test is chosen using the following algorithm: If there are transitions from the current state that have yet to be tested ( $T_0$ ), one such transition is randomly chosen and tested. Else, if there are transitions that have only weakly been tested ( $T_1$ —the ending state has not been reached) a weakly tested transition is randomly chosen and tested. Otherwise, an already-tested transition ( $T_2$ ) is tested. The output from the test is then compared to the specification. This does describe using an algorithm to choose which transition to test next, while the program is running. However, this does not teach or suggest “*identified program behavior*.” The next transition is chosen depending on whether it has yet to be tested, not “program behavior” as the specific program behavior that is being tested in Lee is incidental to the test being performed, and as

far as the applicants can tell, is not mentioned. This also does not teach or suggest a “*state representing the identified program behavior.*” The program behavior of the states in Lee, as far as applicant can tell, is not used within the systems and methods taught by Lee. For example, Fig. 5 and its associated description just mentions whether or not various states have been tested, not what the states represent. This passage of Lee also does not teach or suggest *a selected edge has a **highest probability** of reaching a state representing the identified program behavior.* It does not discuss probabilities of reaching a state.

Lee, 12:4-14. This passage in Lee describes a testing program getting into an unproductive, inadvertent, loop of unknown size. The loop is produced by the testing program choosing a random set of transitions following the same protocol discussed with reference to Fig. 5. However, this does not teach or suggest “*identified program behavior.*” The next transition is chosen depending on whether it has yet to be tested, not “program behavior” as the specific program behavior that is being tested in Lee is incidental to the test being performed; and as far as the applicants can tell, is not mentioned. This also does not teach or suggest a “*state representing the identified program behavior.*” The program behavior of the states in Lee, as far as applicant can tell, is not used within the systems and methods taught by Lee. This passage of Lee also does not teach or suggest *a selected edge has a **highest probability** of reaching a state representing the identified program behavior.* It does not discuss probabilities of reaching a state.

Lee, 6:50-55. Under 35 USC § 102, every limitation of a claim must identically appear in a prior art reference for it to anticipate the claim. The Examiner must not cobble together disparate elements from separate embodiments to make such a rejection. The reference “must clearly and unequivocally disclose the claimed [invention] or direct those skilled in the art to the [invention] without any need for picking, choosing, and combining various disclosures not directly related to each other by the teachings of the cited reference.” [In re Arkley, 455 F.d 586, 587, 175 USPQ 524, 526 (CCPA 1972).] This passage of Lee describes finding reachable global states within a Finite state machine, generally. However, the Method for Conformance Testing which the Action uses to make the §102 rejection does not make such a reachable states calculation. Lee explicitly states “Because we do not do a reachability computation, we cannot, of course say absolutely whether a current global state  $s_i$  can be reached from the previous one  $s_{i-1}$ .” [Lee, 11:20-23.]

As the Examiner has picked and chosen from two separate embodiments to make this §102

rejection, Applicants respectfully submit that the Action has failed to make a *prima facie* case against former claim 19, current claim 15. As such, Applicants respectfully submit that the rejection of claim 37 under 35 U.S.C. § 102(b) is improper and claim 15 in its present form should be patentable. For at least this reason, claim 15 is in condition for allowance.

Further the cited passage does not teach or suggest *wherein determined strategies are determined based on a comparison of edges exiting a deterministic state representing program behavior, and a selected edge has a highest probability of reaching a state representing the identified program behavior.* Generally discussing finding a reachable FSM does not teach or suggest “*identified program behavior.*” The reachable FSM finds *reachable* program states, not *identified* program behavior. This passage of Lee also does not teach or suggest *a selected edge has a **highest probability** of reaching a state representing the identified program behavior.* It does not discuss probabilities of reaching a state, let alone the highest probability of reaching a state.

For all the reasons given above, at least, claim 15 is in condition for allowance.

### **Claims 18 and 20.**

Claims 18 and 20 ultimately depend on claim 15. Thus, at least for the reasons set forth above with respect to claim 15, claims 18 and 20 should also be in condition for allowance. These claims also set forth independently patentable combinations of method acts

### **Request for Interview**

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants’ position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

**Conclusion**

The claims should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By           /Genie Lyons/            
Genie Lyons  
Registration No. 43,841